Table of Contents

[1 Document management history 5](#_Toc454490221)

[1.1 Prepared by 5](#_Toc454490222)

[1.2 Reviewed by 5](#_Toc454490223)

[1.3 Approved by 5](#_Toc454490224)

[1.4 Revision History 1](#_Toc454490225)

[2 Introduction 2](#_Toc454490226)

[3 Purpose of this document 3](#_Toc454490227)

[3.1 Project Scope 3](#_Toc454490228)

[3.2 Definitions, Acronyms and Abbreviations 3](#_Toc454490229)

[3.3 References 4](#_Toc454490230)

[3.4 Overview 4](#_Toc454490231)

[3.5 Business Opportunity 4](#_Toc454490232)

[3.6 Assumptions 4](#_Toc454490233)

[3.7 Risks 4](#_Toc454490234)

[4 Implementation 5](#_Toc454490235)

[4.1 Design Scope 5](#_Toc454490236)

[4.2 Dependencies/Impacts 5](#_Toc454490237)

[4.3 Development Tools 5](#_Toc454490238)

[4.4 Platforms 5](#_Toc454490239)

[4.5 Logical Architecture 6](#_Toc454490240)

[4.6 Architecture Decomposition 6](#_Toc454490241)

[5 Structural (Static Diagrams) 7](#_Toc454490242)

[5.1 Class Diagram 7](#_Toc454490243)

[5.2 Component Diagrams 8](#_Toc454490244)

[5.2.1 Mobile Client Application Components (iOS) 8](#_Toc454490245)

[5.2.2 Mobile Client Application Components (Android) 9](#_Toc454490246)

[5.3 Activity/Behavioral Diagrams 9](#_Toc454490247)

[5.3.1 Overall (General) Application Process 9](#_Toc454490248)

[5.4 Sequence Diagrams – Rider/Driver 10](#_Toc454490249)

[5.4.1 Registration 10](#_Toc454490250)

[5.4.2 Login 10](#_Toc454490251)

[5.4.3 Accept EULA 10](#_Toc454490252)

[5.4.4 Profile 11](#_Toc454490253)

[5.4.5 Card info 11](#_Toc454490254)

[5.4.6 Last Trip 11](#_Toc454490255)

[5.4.7 Report a problem 11](#_Toc454490256)

[5.4.8 Contact 11](#_Toc454490257)

[5.5 Sequence Diagrams – Rider 11](#_Toc454490258)

[5.5.1 Request ride(now) 11](#_Toc454490259)

[5.5.2 Schedule ride 11](#_Toc454490260)

[5.6 Sequence Diagrams – Driver 12](#_Toc454490261)

[5.6.1 Vehicle details 12](#_Toc454490262)

[5.6.2 License details 12](#_Toc454490263)

[5.6.3 Insurance details 12](#_Toc454490264)

[5.6.4 Accept ride request 12](#_Toc454490265)

[5.6.5 Arrive Pickup location 12](#_Toc454490266)

[5.6.6 Start Ride 12](#_Toc454490267)

[5.6.7 End ride 12](#_Toc454490268)

[5.6.8 Go Online 12](#_Toc454490269)

[5.6.9 Go Offline 13](#_Toc454490270)

[5.6.10 Bank info 13](#_Toc454490271)

[5.6.11 Ride History 13](#_Toc454490272)

[5.6.12 Settings 13](#_Toc454490273)

[5.6.13 Request Ride 13](#_Toc454490274)

[5.6.14 On location change 13](#_Toc454490275)

[6 Layering 14](#_Toc454490276)

[6.1 Mobile App Presentation Layer 14](#_Toc454490277)

[6.1.1 Mobile Client Hybrid Application Screens 14](#_Toc454490278)

[6.1.2 Web Admin Application Screens 20](#_Toc454490279)

[6.1.3 Web Site for end users 20](#_Toc454490280)

[6.2 Data Formats 20](#_Toc454490281)

[6.2.1 REST API data request/ response 20](#_Toc454490282)

[6.3 Database Tables 20](#_Toc454490283)

[7 Services 22](#_Toc454490284)

[7.1 Service Methods 22](#_Toc454490285)

[7.1.1 Service Methods 22](#_Toc454490286)

[8 Security 25](#_Toc454490287)

[8.1 Authentication 25](#_Toc454490288)

[8.1.1 Mobile Authentication 25](#_Toc454490289)

[8.2 Authorization 25](#_Toc454490290)

[8.2.1 NA 25](#_Toc454490291)

[8.3 Database Connections 25](#_Toc454490292)

[8.4 Data protection 25](#_Toc454490293)

[8.5 Other considerations 25](#_Toc454490294)

[8.5.1 Session Management 25](#_Toc454490295)

[8.5.2 Cryptography 25](#_Toc454490296)

[8.5.3 Secure communications 25](#_Toc454490297)

[8.5.4 Data confidentiality 25](#_Toc454490298)

[8.5.5 System Design and Architecture 25](#_Toc454490299)

[8.5.6 System Configuration 25](#_Toc454490300)

[8.5.7 Secure Development 25](#_Toc454490301)

[9 Performance and Robustness 27](#_Toc454490302)

[9.1 Memory Management 27](#_Toc454490303)

[9.2 Processes and threads 27](#_Toc454490304)

[9.3 Concurrency 27](#_Toc454490305)

[9.4 Transactions 27](#_Toc454490306)

[9.5 Exception/Error handling 27](#_Toc454490307)

[9.6 Localization/Internationalization 27](#_Toc454490308)

[9.7 Benchmarking 27](#_Toc454490309)

[10 Configuration Management 28](#_Toc454490310)

[10.1 Source Control 28](#_Toc454490311)

[10.2 Configuration Files 28](#_Toc454490312)

[10.3 Initial Configuration 28](#_Toc454490313)

[10.3.1 Initial Packing of the Application 28](#_Toc454490314)

[10.4 Configuration Requirements 28](#_Toc454490315)

[10.5 Server Requirements 28](#_Toc454490316)

[11 Other Design Considerations 29](#_Toc454490317)

[11.1 Reporting 29](#_Toc454490318)

[11.2 Scheduling Impacts 29](#_Toc454490319)

[11.3 Non-Functional Requirements 29](#_Toc454490320)

[11.4 Backup 29](#_Toc454490321)

[11.5 Disaster Recovery 29](#_Toc454490322)

[11.6 Archive and Purge Strategy 29](#_Toc454490323)

[11.7 Installer 29](#_Toc454490324)

[11.8 Upgrade Strategy 29](#_Toc454490325)

[11.9 Data Conversion 29](#_Toc454490326)

[11.10 Metrics 29](#_Toc454490327)

[12 Appendix 30](#_Toc454490328)

# Document management history

## Prepared by

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Santhosh Joseph | Solution Architect | Intimation | 15/06/16 |
| Athira Satheesh | Developer | Intimation |  |
| Bibin Baby | Developer | Intimation |  |
| Arya M | Developer | Intimation |  |

## Reviewed by

The following section describes who has reviewed this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Shenjin Thomas | Client | BZRide inc | Xx/xx/xx |
|  |  |  |  |
|  |  |  |  |

## Approved by

The following section describes who has approved this document:

|  |  |  |  |
| --- | --- | --- | --- |
| Name | Title | Organization | Date |
| Shenjin Thomas | Client | BZRide inc | Xx/xx/xx |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

## 

## Revision History

The following section describes the revision history:

|  |  |  |  |
| --- | --- | --- | --- |
| Version | Date | Author | Comments |
| 0.1 | 15/06/2016 | Santhosh | Initial Draft |
| 1.0 | 16/06/2016 | Santhosh | Updated for different sections and diagrams |
| 1.1 | 22/06/2016 | Santhosh | Service methods and tables |
| 1.2 | 23/06/2016 | Santhosh | DB table are added, removed unwanted sections |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |
|  |  |  |  |

# Introduction

The introduction of the Solution Design Document provides an overview of the entire document. It includes the purpose, scope, definitions and references of this document.

# Purpose of this document

This document provides an overview of the BZRide Mobile Application. This document is intended to capture and convey the significant architectural and design decisions which have been made for designing and building the application. This is a technical reference document for team members involved in the development, testing and implementation of the application.

## Project Scope

BZRide is for driver and riders. The app and admin system and web site are developed as part of this project.

## Definitions, Acronyms and Abbreviations

|  |  |
| --- | --- |
| Term / acronym / abbreviation | Definition |
| REST | Representational State Transfer |
| API | Application Programming Interface |
| JSON | JavaScript Object Notation |
| AJAX | Asynchronous JavaScript and XML |
| CSS | Cascaded Style Sheet |
| WCF | Windows Communication Foundation |
| EULA | End User License Agreement |

## References

This section provides a complete list of documents referenced in this document. Each document is identified by title, report number (if applicable), date, and publishing organization. Specify the sources from where the references can be obtained. This information may be provided by reference to an appendix or to another document.

This section contains a listing of existing documents that were referenced in order to prepare this document.

* High level Requirements document
* [SAD](https://share.ey.net/sites/tassdprojects/das/Shared%20Documents/Architecture/DAS%20Mobile%20App%20Solution%20Architecture%20Document.docx?Web=1)
* [Wireframe/VD](https://share.ey.net/sites/tassdprojects/das/Shared%20Documents/Architecture/Wireframes/DAS-v2.3.zip)
* Non Functional Requirements if any

## Overview

Riders want to hire taxi can make ride request and the same will be routed to drivers available near the location. Driver accepts the request and reaches the pickup location and start ride with the help of app navigation. Finally the driver close the ride and rider get charged with the fare calculated based on distance travelled and time.

## Business Opportunity

* Intimation can get more clients and projects as we get good feedback on the success of the BZRide application.

## Assumptions

For the purposes of designing and developing the BZRide Mobile application, the following assumptions have been made by the application development team:

* The mobile client application would be developed as an native application.
* The application would be available only for iPhones and Android smart phones.
* The APIs and credentials for Stripe and Braintree will be provided by client
* The production server will be procured by client
* REST APIs would be available for accessing all data from php/mysql back end.
* None of the card / bank information will be stored in back end system as such. Instead encrypted tokens will be used.

## Risks

* None.

# Implementation

## Design Scope

The BZRide Mobile application has been broken down into three major subsystems, the native applications, Admin web site /public web site for end users (driver/rider) and a set of REST APIs that are running on web portal.

## Dependencies/Impacts

Describe any dependencies and/or downstream impacts

## Development Tools

| Area of Development Use | Tool / Technology Currently Utilized |
| --- | --- |
| Integrated Development Environment | Dreamweaver, WebStorm, PHPStorm, XCode 7.0 |
| Development Framework | iOS SDK 9.0, Android SDK |
| Primary Development Languages | JAB, PHP, Objective C, IOS/Android SDK, JAVA HTML, CSS |
| Mobile Operating Systems | iOS 7.1 or Higher (iPhone) |
| Mobile Bus Width | iPhone 64 bit processing |
| User Experience Components | HTML5, CSS3 |
| Web Browser Platforms | WebKit enabled browsers |

## Platforms

Mobile App Development is planned in native languages and SDKs. Data will be pulled from MYSQL DB using REST API calls.

## Logical Architecture

The logical architecture of the solution is shown here:

## Architecture Decomposition



# Structural (Static Diagrams)

## Class Diagram

The Class Diagram of the solution is as shown below:

TODO

## Component Diagrams

TBD

### Mobile Client Application Components (iOS)

Various application components of BZRide Mobile iPhone application are,

* Native classes in Objective C

These sections are explained in detail in the Application Design section.

#### Constants.js

Shared variables and constant functions are defined in this JavaScript file.

#### Login.js

Other supporting components are explained below.

### Mobile Client Application Components (Android)

#### Constants.js

Shared variables and constant functions are defined in this JavaScript file.

#### Login.js

#### Webservice.js

Makes the web service calls. Acts as a wrapper for other javascript files to get web service data.

This class can be further split into different service classes.

#### Utils.js

A common javascript class for holding common utility functions across all other classes.

## Activity/Behavioral Diagrams

### Overall (General) Application Process

The overall application process is depicted below.

Authentication Process

The User Authentication process of the application is as the following



## Sequence Diagrams – Rider/Driver

### Registration

Accept the details for registration on Application screen.

Accept credit/debit card info. Send card info to Stripe API(https://stripe.com/) for making it as an encrypted token

If rider pass all the details to web service method **RegisterRider** along with above card token.

If driver pass all the details to web service method **RegisterDriver** along with above card token.

### Login

Accept the login details on Application screen.

If rider Pass email and password to web service method **LoginRider**.

If driver Pass email and password to web service method **LoginDriver**.

### Accept EULA

After registration screen App moves to License screen.

On Accept call web service **AcceptEULARider or**  **AcceptEULADriver** and on server side mark LicenseAccepted flag as true.

On Reject show warning message “You cannot continue unless end user license agreement is accepted”

### Profile

Show Profile details on Application screen(**GetRiderProfile** or **GetDriverProfile** method). Edit the details if required. Once user click Save, call the web service method **UpdateDriverProfile** if driver. If rider call **UpdateRiderProfile**.

### Card info

Show card details on Application screen (**GetCardDetails** method). Edit the details if required. Once user click Save, invoke stripe API to get token and call the web service method **UpdateCardDetails** Method.

### Last Trip

Show very latest trip details by calling web service method **GetLastRideDetails.** If no data found show message “No Ride data found”

### Report a problem

Show an option text one paragraph for problem description and one title. On Send button save to server using service method **ReportProblem**

### Contact

Show contact Info as email and phone number. [info@bzride.com](mailto:info@bzride.com) and (xxx)-(xxx)-(xxxx)

## Sequence Diagrams – Rider

### Request ride(now)

Tap Request Now button from App screen.

Select Destination

Tap Confirm button.

Create a ride request with start location and destination location (lat/long) and address riderid etc.

Call web service method **CreateRideRequestImmediate**

### Schedule ride

Tap Request Now button from App screen.

Select Destination

Select Date and Time of travel (validate if it is greater than 2 days ahead of current date. Also warn if date is previous)

Tap Confirm button.

Create a ride request with start location and destination location (lat/long) trip date and time, and address riderid etc.

Call web service method **CreateRideRequestSchedule**

## Sequence Diagrams – Driver

### Vehicle details

The details will be included in **GetDriverProfile** service method as inside tag vehicle

### License details

The details will be included in **GetDriverProfile** service method as inside tag license

### Insurance details

The details will be included in **GetDriverProfile** service method as inside tag insurance

### Accept ride request

The ride request is accepted as a push notification.

Handle the event in push notification handler for ios/android. Contains id, requestor name, start, destination

Call web service **ReadRideRequest**(id). If previous request is trying to be read show alert.

Driver accept it by tapping accept button within 20 seconds call web service method **AcceptRideRequest**(id,driverid)

Note: Back end service will monitor the accept status from any driver and if not accepted by anybody, give the message to requestor.

### Arrive Pickup location

The ride request is accepted as a push notification. Just drive to the pick location given by rider.

### Start Ride

The web service call **StartRide**(id) given by rider. Ride request table updated with status as riding-**R**

And driver table updated with status as driving-**D**. Open map view with start and end locations and get into start navigation mode as well. Use google maps view or waze View depending on settings menu option

### End ride

The web service call **EndRide**(id) given by rider. Ride request table updated with status as complete-**C**

And driver table updated with status as Available-**A.**

End navigation mode.

Calculate distance travelled.

Calculate time taken

### Go Online

The web service call **UpdateDriverAvailability**(id,true) to update status. Available-**A**

### Go Offline

The web service call **UpdateDriverAvailability**(id,false) to update status. Available-**O**

### Bank info

The **GetBankInfo** service method on updating details call method **UpdatebankInfo**

### Ride History

The **GetRideHistory** service method for pulling details

### Settings

Show a page with two radio buttons google map and waze and set the value G or W (1 or 2 also ) in local app settings

### Request Ride

Same work flow as Rider do Request

### On location change

On location change call service method **UpdateDriverLocation**(id,lat,long)

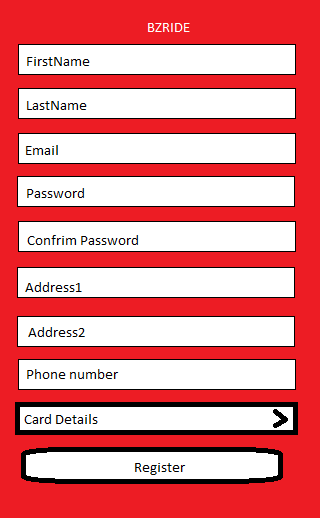
Layering

Layering represents an ordered grouping of functionality, with the application-specific located in the upper layers, functionality that spans application domains in the middle layers, and functionality specific to the deployment environment at the lower layers

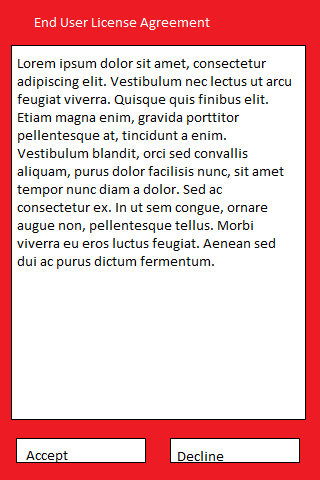
## Mobile App Presentation Layer

### Mobile Client Hybrid Application Screens

#### Register User



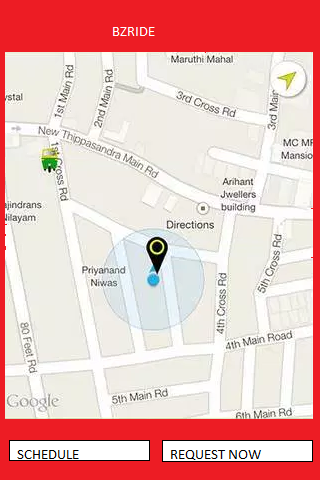
#### EULA Page



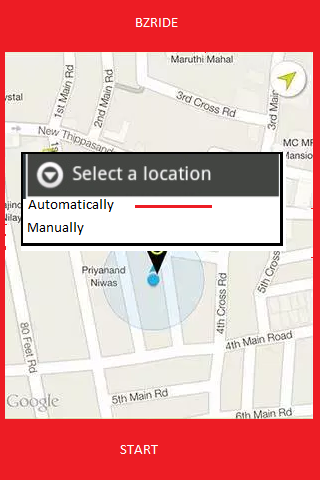
#### login Screen



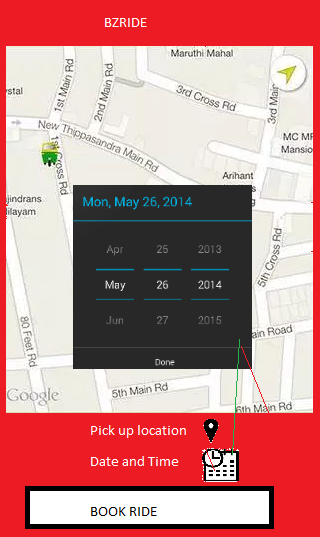
#### Home Page



#### Home Page-Pick Up location



#### Home Page-Schedule Trip



### Web Admin Application Screens

#### Admin Pages for listing drivers and for Riders and checkbox for activate/deactivate

### Web Site for end users

Paste designed site pages here

## Data Formats

### REST API data request/ response

Write basic methods and paste data request/response format here

## Database Tables

|  |  |  |
| --- | --- | --- |
| TableName | columns | Comments if any |
| bztblRiders | Id,FirstName,LastName,Email,Password,Address1,Address2,Phone  isLicenseAccepted, isActive,  CardType(C/D), CardProvider(M/V), CardToken,  CreatedByDate |  |
| bztblDrivers | Id,FirstName,LastName,Email,Password,Address1,Address2,Phone  isLicenseAccepted, isActive,status(A,O,D),  currentlat,currentlong,  CreatedByDate |  |
| bztblDriverLicenseDetails | Id,driverid,licNumber,licStateIssued,licDateIssued,licExpDate, CreatedByDate |  |
| bztblDriverVehicleDetails | Id,Driverid, VehicleModel,VMake,VColor,VYear,  VehicleNumber,VRegState,VDateRegistered,VExpDate,  CreatedByDate |  |
| bztblUserBankDetails | Id,userid,Type,BankName,Account Number,AccountHoldername,Routing Number, CreatedByDate | (encrypt except id, Type,bank name, CreatedByDate) |
| bztblDriverinsuranceDetails | Id,driverid, InsCompany,InsPolicyNumber,InsValidFrom,InsExpDate, CreatedByDate |  |
| bztblRideRequests | id,requesttype(immediate- I, Future-F)requestorid,driverid,startlocation,endlocation,startlat,startlong,endlat,endlong, status(A,R,C) , rideDate, rideTime,actual rideDate, actualrideTime, CreatedByDate |  |
| bztblReportedProblems | Id,reporttitle,reportdescription, CreatedByDate |  |
|  |  |  |
|  |  |  |

Services

BZRide back end system exposes REST methods to GET/POST data to the system. Major services are below

|  |  |
| --- | --- |
| Service URL | Feature Affected |
| www.intimationsoftware.com/ws/bzride(QA) | All |
| www.bzride.com/ws/UAT/bzride(Staging) | All |
| www.bzride.com/ws/Prod/bzride(Production) | All |
|  |  |
|  |  |
|  |  |
|  |  |

## Service Methods

Below section covers all web service methods for BZRide application.

### Service Methods

|  |  |  |
| --- | --- | --- |
| Service methods | Parameters | Returns |
| RegisterRider | FirstName,LastName,Email,Password,Address1,Address2,Phone  CardType(C/D), CardProvider(M/V), CardToken | S/F |
| RegisterDriver | FirstName,LastName,Email,Password,Address1,Address2,Phone  InsCompany,InsPolicyNumber,InsValidFrom,InsExpDate, licNumber,licStateIssued,licDateIssued,licExpDate, VehicleModel,VMake,VColor,VYear,  VehicleNumber,VRegState,VDateRegistered,VExpDate | S/F |
| LoginRider | Email,Password | S/F |
| LoginDriver | Email,Password | S/F |
| AcceptEULARider | id,accptFlag | S/F |
| AcceptEULADriver | id,accptFlag | S/F |
| GetRiderProfile | id | S/F, Rider Details |
| GetDriverProfile | id | S/F, Driver Details |
| UpdateDriverProfile | id,profile fields | S/F |
| UpdateRiderProfile | id,profile fields | S/F |
| GetCardDetails | id | S/F, Card Details |
| UpdateCardDetails | id,CardType(C/D),CardProvider(M/V),CardToken | S/F |
| GetLastRideDetails | Id(rider) | S/F, Ride details |
| ReportProblem | id, reporttitle, reportdescription | S/F |
| CreateRideRequest | id,requestorid,startlocation,endlocation,startlat,startlong,endlat,endlong,RideType(I,F) | S/F |
| ReadRideRequest | requestid | S/F, Ride Details |
| AcceptRideRequest | requestid,driverid | S/F |
| StartRide | requestid | S/F |
| EndRide | requestid | S/F |
| UpdateDriverAvailability | driverid,flag | S/F |
| GetBankInfo | id | S/F, Bank Details |
| GetRideHistory | driverid | S/F, Ride details |
| UpdateDriverLocation | driverid,lat,long | S/F |
| UpdateBankInfo | id,BankName,Account Number,AccountHoldername,Routing Number(encrypt except id, bank name) | S/F |

Security

## Authentication

The different Authentication models used are as follows:

### Mobile Authentication

## Authorization

### NA

## Database Connections

NA

## Data protection

NA

## Other considerations

### Session Management

### Cryptography

### Secure communications

1. All server communications are secured with SSL.??

### Data confidentiality

1. No data is stored on device and the App shows everything dynamically by calling REST APIs.
2. The course information/discussion threads are not of confidential nature.

### System Design and Architecture

### System Configuration

1. Mobile application uses REST API for data downloading.

### Secure Development

1. User credentials such as passwords, passphrases, PINs and digital certificates (PKI) are not stored anywhere in the mobile device.
2. Application does not allow sharing of its files and documents with unapproved Mobile Applications on the same mobile device.
3. Application restricts copy and paste operations so its clipboard information cannot be shared with unapproved Mobile Applications.
4. Application restricts its application data from being copied to connected computer-based or cloud-based backup services.

Performance and Robustness

## Memory Management

Application will load the data using REST API and it is kept in device memory for rendering the UI. Profiling using chrome can be done for better check on performance and memory management.

## Processes and threads

NA

## Concurrency

NA

## Transactions

## Exception/Error handling

Default exception/error handling methods provided by iOS/Android/Javascript will be used.

Details of the errors that the php/mysql application encounters are displayed to the user on the web page. The errors encountered are also logged.

## Localization/Internationalization

NA

## Benchmarking

NA

Configuration Management

For detailed information on management configuration, include section if applicable.

## Source Control

App source files are committed to GitHub.

## Configuration Files

NA

## Initial Configuration

### Initial Packing of the Application

The App will be bundled as ipa or apk file and uploaded to store.

## Configuration Requirements

NA

## Server Requirements

NA.

# Other Design Considerations

These sections are optional and are included only if applicable.

## Reporting

NA

## Scheduling Impacts

NA

## Non-Functional Requirements

## Backup

NA

## Disaster Recovery

NA

## Archive and Purge Strategy

NA

## Installer

NA

## Upgrade Strategy

NA

## Data Conversion

NA

## Metrics

NA

# Appendix

NIL